This document summarizes Siyuan Tang's exposure to and understanding of research areas of interest. The application system does not provide a section to submit such a document.

Understanding of Research

Siyuan Tang

Last modified on January 3, 2025.

"All models are wrong, but some are useful." (Box and Draper (1987))

Contents

1 Research Internship at the University of Michigan

1.1 Research Experience in Ann Arbor

I was an on-site research intern working in Professor Yang Chen's group at the Department of Statistics, University of Michigan (UM), Ann Arbor, Michigan, from July 2024 to November 2024.

Our work focused on predicting the energy released by solar flares and utilizing this information to classify flares into pre-defined categories. Our focus extended beyond regression performance to include binary classification performance, measured by the True Skill Statistic (TSS; Woodcock (1976)). Early on in this project I encountered a challenge: some solar flares had ill-defined and extremely long durations, complicating the analysis of their dynamics. To address this, I applied a truncation method, focusing on the critical moments surrounding the peak times of the flares, enabling more accurate modeling of their dynamics. I then proposed a profile modeling strategy to effectively denoise the observed sequences by employing a trend function that assumed linear growth before the peak and linear decay after. I fitted the model and conducted thorough goodness-of-fit assessments by evaluating residual sequences and testing for independence and normality.

Additionally, I explored how energy release patterns could enhance the classification of flare events, especially for stronger ones. Our work marked the first statistical analyses of solar flares from an energy perspective and illustrated that energy-based classifications often outperform traditional intensity-based methods (Jiao et al. (2020)). Currently, the team is preparing a paper that summarizes our main findings.

1.2 Working Remotely After Returning to China

I continued to work after coming back to China in November 2024. My research focused on developing robust binary classification methods that perform well under distribution shift, particularly for solar flare prediction tasks. I explored two types of distribution shift:

- Label shift, where the proportion of positive cases varies between training and test data.
- Covariate shift, where the feature distribution changes.

For label shift, I proved that TSS optimization is inherently invariant to changes in class proportions, and derived the closed form of the optimal classifier that maximizes TSS. To overcome the discontinuity of TSS, I developed a smooth approximation using sigmoid functions. My theoretical results were verified on synthetic data. Through real data experiments on the GOES event list (Garcia (1994)), I demonstrated that TSS optimization maintains robust performance under varying degrees of label shift, often outperforming traditional classifiers like Support Vector Machines (Cortes (1995)), Linear Discriminant Analysis, and Quadratic Discriminant Analysis.

For covariate shift, I implemented an importance weighting strategy to correct for the distribution difference between training and test data, deriving a modified TSS objective that incorporates these weights. This methodology was inspired by the textbook Sugiyama and Kawanabe (2012) and recent work from Professor Tianxi Cai's group Wang et al. (2024). However, the results were not better than those for label shift, suggesting that the label shift assumption may be more appropriate for solar flare prediction.

1.3 Lessons from the Internship

This research experience has not only deepened my understanding of multivariate methods but also broadened my perspective on real-world problem-solving. Specifically, I have gained the following lessons:

Real Data. Distribution shifts, misclassified instances, missing values, and weak signals (low signal-to-noise ratios) are commonly encountered in real-world data, which is totally different from "textbook examples".

Attitudes toward Existing Literature. Everyone makes mistakes. Good researchers should not completely trust the results from a paper without careful thought. It is sometimes necessary to verify the claims through hands-on implementation.

Simulation Studies. When faced with challenges or counterintuitive results in real-data analysis, conducting simulation studies with synthetic data can often provide valuable insights and validate the reasoning.

Communication and Cooperation. Cooperation is essential in modern research. Good researchers should always save their collaborators' time and communicate effectively. For example, preparing a brief document summarizing key points from the previous meeting, recent progress, and current questions can greatly facilitate the discussion—this is a habit I have maintained.

Academic Writing. In formal writing, it is important to avoid vague statements such as "almost", "many (much)", "most", "roughly", and "approximately". Instead, we can use precise numbers to convey clear and accurate information. Additionally, taking high-quality notes can help clarify the challenges we are facing and the tools at our disposal. Summarizing the ideas in organized blocks (which is a habit I have maintained) can also save time and improve efficiency in academic writing.

2 Statistical Network Analysis

Preliminaries. We denote a network with n nodes by its adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, where $A_{ij} = 1$ if there is an edge (link) between nodes i and j and $A_{ij} = 0$ otherwise. We do not allow for self-edges (self-loops), i.e., the diagonal entries of $\boldsymbol{A}$ are all zero.

2.1 The Stochastic Block Model

Among various models, the stochastic block model (SBM; Holland et al. (1983)) has attracted much attention and is arguably the best studied and most commonly used. Below, we briefly review its setting.

Suppose there are K communities. Each node belongs to only one of the communities. Let $\boldsymbol{c} = (c_1, \ldots, c_n) \in \{1, \ldots, K\}^n$ denote the true community labels of the nodes, and assume c_i's are i.i.d. categorical variables with parameter vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, where $\pi_k \geq 0$ for all $1 \leq k \leq K$ and $\sum_{k=1}^{K} \pi_k = 1$. Conditional on the community labels, the edge variables A_{ij}'s are independent Bernoulli variables with $\mathbb{E}[A_{ij} \mid \boldsymbol{c}] = P(A_{ij} = 1 \mid \boldsymbol{c}) = P_{c_i c_j}$, where $\boldsymbol{P} \in [0,1]^{K \times K}$ is the symmetric edge-probability matrix with the kl-entry P_{kl} characterizing the probability of connection between nodes in communities k and l. Let $\boldsymbol{\Omega} = (\boldsymbol{\pi}, \boldsymbol{P})$.

2.1.1 Why Estimation via EM is Intractable

It would be natural and logical to treat $\boldsymbol{c}$ as latent variables and attempt to fit the SBM using EM algorithm. The log-likelihood for complete data (i.e., $\boldsymbol{A}$ and $\boldsymbol{c}$) is computed as

$$
\begin{aligned}
\ell(\boldsymbol{\Omega}; \boldsymbol{A}, \boldsymbol{c}) &= \log P(\boldsymbol{A}, \boldsymbol{c} \mid \boldsymbol{\Omega}) && (1)\\
&= \log P(\boldsymbol{c} \mid \boldsymbol{\pi}) + \log P(\boldsymbol{A} \mid \boldsymbol{c}, \boldsymbol{P}) && (2)\\
&= \sum_{i=1}^{n} \log P(c_i \mid \boldsymbol{\pi}) + \sum_{i<j} \log P(A_{ij} \mid c_i, c_j, \boldsymbol{P}) && (3)\\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} I(c_i = k) \log \pi_k && (4)\\
&\quad + \sum_{i<j} \left[A_{ij} \log P(A_{ij} = 1 \mid c_i, c_j, \boldsymbol{P}) + (1 - A_{ij}) \log P(A_{ij} = 0 \mid c_i, c_j, \boldsymbol{P})\right] && (5)\\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} I(c_i = k) \log \pi_k && (6)\\
&\quad + \sum_{i<j} \sum_{k=1}^{K} \sum_{l=1}^{K} I(c_i = k) I(c_j = l) \left[A_{ij} \log P_{kl} + (1 - A_{ij}) \log (1 - P_{kl})\right]. && (7)
\end{aligned}
$$

In order to perform EM algorithm, we first need to derive the posterior distribution of $\boldsymbol{c}$, given the observed network $\boldsymbol{A}$ and the model parameters $\boldsymbol{\Omega}$. Specifically, we have

$$
\begin{aligned}
P(\boldsymbol{c} \mid \boldsymbol{A}, \boldsymbol{\Omega}) &= \frac{P(\boldsymbol{c}, \boldsymbol{A} \mid \boldsymbol{\Omega})}{P(\boldsymbol{A} \mid \boldsymbol{\Omega})} && (8)\\
&\propto P(\boldsymbol{c}, \boldsymbol{A} \mid \boldsymbol{\Omega}) && (9)\\
&= P(\boldsymbol{c} \mid \boldsymbol{\pi}) P(\boldsymbol{A} \mid \boldsymbol{c}, \boldsymbol{P}) && (10)\\
&= \prod_{i=1}^{n} \pi_{c_i} \prod_{i<j} \left(P_{c_i c_j}\right)^{A_{ij}} \left(1 - P_{c_i c_j}\right)^{1 - A_{ij}}, && (11)
\end{aligned}
$$

which leads to the following updating formula:

$$\pi_k^{(t+1)} \propto \sum_{i=1}^{n} P(c_i = k \mid \boldsymbol{A}, \boldsymbol{\Omega}^{(t)}), \quad 1 \leq k \leq K, \tag{12}$$

$$P_{kl}^{(t+1)} \propto \sum_{i<j} P(c_i = k, c_j = l \mid \boldsymbol{A}, \boldsymbol{\Omega}^{(t)})\ A_{ij}, \quad 1 \leq k, l \leq K. \tag{13}$$

We note, however, that the posterior distribution of $\boldsymbol{c}$ given by Eq. (11) is intractable for large n and K, since it is of order $\mathcal{O}(K^n)$. It is also computationally infeasible to solve for $\arg\max_{\boldsymbol{c}} P(\boldsymbol{c} \mid \boldsymbol{A}, \boldsymbol{\Omega})$, which poses a great challenge for community detection.

2.2 A Pseudo-Likelihood Method

Wang et al. (2023) proposed a pseudo-likelihood-based method[1] which performs well for both small and large scale networks. Here we describe their method for fitting the SBM.

We introduce an initial column labeling vector $\boldsymbol{e} = (e_1, \ldots, e_n) \in \{1, \ldots, K\}^n$. Let $\boldsymbol{a}_i = (A_{i1}, \ldots, A_{in})$ denote the i-th row of $\boldsymbol{A}$, $1 \leq i \leq n$. The pseduo-likelihood function[2] is defined as

$$\begin{aligned}
\mathcal{L}_{\mathrm{PL}}(\boldsymbol{\Omega}, \boldsymbol{e}; \{\boldsymbol{a}_i\}) &= P(\{\boldsymbol{a}_i\} \mid \boldsymbol{\Omega}, \boldsymbol{e}) && \text{(Treat } \boldsymbol{e} \text{ as model parameters)} && (14)\\
&= \prod_{i=1}^{n} P(\boldsymbol{a}_i \mid \boldsymbol{\Omega}, \boldsymbol{e}) && \text{(Ignore the potential dependency)} && (15)\\
&= \prod_{i=1}^{n} \sum_{k=1}^{K} P(\boldsymbol{a}_i, c_i = k \mid \boldsymbol{\Omega}, \boldsymbol{e}) && \text{(Treat } \boldsymbol{c} \text{ as a vector of latent variables)} && (16)\\
&= \prod_{i=1}^{n} \left\{ \sum_{k=1}^{K} P(c_i = k \mid \boldsymbol{\Omega})\ P(\boldsymbol{a}_i \mid c_i = k, \boldsymbol{\Omega}, \boldsymbol{e}) \right\} && && (17)\\
&= \prod_{i=1}^{n} \left\{ \sum_{k=1}^{K} \pi_k \prod_{j=1}^{n} P(A_{ij} \mid c_i = k, \boldsymbol{\Omega}, \boldsymbol{e}) \right\} && && (18)\\
&= \prod_{i=1}^{n} \left\{ \sum_{k=1}^{K} \pi_k \prod_{j=1}^{n} P_{ke_j}^{A_{ij}} \left(1 - P_{ke_j}\right)^{1-A_{ij}} \right\}, && && (19)
\end{aligned}$$

with its logarithm as

$$\ell_{\mathrm{PL}}(\Omega, \boldsymbol{e}; \{\boldsymbol{a}_i\}) = \sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} \pi_k \prod_{j=1}^{n} P_{ke_j}^{A_{ij}} \left(1 - P_{ke_j}\right)^{1-A_{ij}} \right\}. \tag{20}$$

2.2.1 Estimation via Alternating Update

We use an alternating update algorithm to estimate the model parameters $\boldsymbol{\Omega}$ and $\boldsymbol{e}$.

[1] The idea of pseudo-likelihood dates back to Besag (1974) and generally involves ignoring some of the dependency structure in the data to simplify the likelihood and make it more tractable.

[2] In the original paper Wang et al. (2023), this function is referred to as the *profile-pseudo likelihood*, as $\boldsymbol{\Omega}$ is a nuisance parameter while $\boldsymbol{e}$ is the parameter of interest.

Given the current estimate of column labeling vector denoted by $\hat{\boldsymbol{e}}$, maximizing $\ell_{\mathrm{PL}}(\boldsymbol{\Omega}, \hat{\boldsymbol{e}}; \{\boldsymbol{a}_i\})$ over $\boldsymbol{\Omega}$ can be conducted by a vanilla EM algorithm (treating c_i's as latent variables), where each M-step admits a closed form:

$$\pi_k^{(t+1)} \propto \sum_{i=1}^{n} P(c_i = k \mid \boldsymbol{a}_i, \boldsymbol{\Omega}^{(t)}, \hat{\boldsymbol{e}}), \quad 1 \le k \le K, \tag{21}$$

$$P_{kl}^{(t+1)} \propto \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} P(c_i = k \mid \boldsymbol{a}_i, \boldsymbol{\Omega}^{(t)}, \hat{\boldsymbol{e}}) I(\hat{e}_j = l), \quad 1 \le k, l \le K. \tag{22}$$

The quantity $P(c_i = k \mid \boldsymbol{a}_i, \boldsymbol{\Omega}^{(t)}, \hat{\boldsymbol{e}})$ can be computed as

$$\begin{aligned} P(c_i = k \mid \boldsymbol{a}_i, \boldsymbol{\Omega}^{(t)}, \hat{\boldsymbol{e}}) &\propto P(c_i = k, \boldsymbol{a}_i \mid \boldsymbol{\Omega}^{(t)}, \hat{\boldsymbol{e}}) & (23)\\ &= P(c_i = k \mid \boldsymbol{\Omega}^{(t)})\; P(\boldsymbol{a}_i \mid c_i = k, \boldsymbol{\Omega}^{(t)}, \hat{\boldsymbol{e}}) & (24)\\ &= \pi_k^{(t)} \prod_{j=1}^{n} \left(P_{k\hat{e}_j}^{(t)}\right)^{A_{ij}} \left(1 - P_{k\hat{e}_j}^{(t)}\right)^{1-A_{ij}}. & (25) \end{aligned}$$

Given the current estimate of $\boldsymbol{\Omega}$ denoted by $\widehat{\boldsymbol{\Omega}}$, maximizing $\ell_{\mathrm{PL}}(\widehat{\boldsymbol{\Omega}}, \boldsymbol{e}; \{\boldsymbol{a}_i\})$ over all possible $\boldsymbol{e}$'s is of order $\mathcal{O}(K^n)$, which is in fact intractable. In order to efficiently update $\boldsymbol{e}$, Wang et al. (2023) proposed the following updating formula:

$$e_j^{(s+1)} \leftarrow \underset{k \in \{1,\dots,K\}}{\arg\max} \sum_{i=1}^{n} \sum_{l=1}^{K} P(c_i = l \mid \boldsymbol{a}_i, \widehat{\boldsymbol{\Omega}}, \boldsymbol{e}^{(s)}) \left\{ A_{ij} \log \hat{P}_{lk} + (1 - A_{ij}) \log(1 - \hat{P}_{lk}) \right\}, \tag{26}$$

where the update for $\boldsymbol{e}$ is obtained separately for each node. Eq. (26) can be intuitively justified as follows:

$$\begin{aligned} &\sum_{i=1}^{n} \sum_{l=1}^{K} P(c_i = l \mid \boldsymbol{a}_i, \widehat{\boldsymbol{\Omega}}, \boldsymbol{e}^{(s)}) \left\{ A_{ij} \log \hat{P}_{le_j^{(s+1)}} + (1 - A_{ij}) \log(1 - \hat{P}_{le_j^{(s+1)}}) \right\} & (27)\\ =&\sum_{i=1}^{n} \sum_{l=1}^{K} P(c_i = l \mid \boldsymbol{a}_i, \widehat{\boldsymbol{\Omega}}, \boldsymbol{e}^{(s)}) \log P(A_{ij} \mid c_i = l, e_j^{(s+1)}, \widehat{\boldsymbol{\Omega}}) & (28)\\ =&\sum_{i=1}^{n} \mathbb{E}_{c_i \mid \boldsymbol{a}_i, \widehat{\boldsymbol{\Omega}}, \boldsymbol{e}^{(s)}} \left[\log P(A_{ij} \mid c_i, e_j^{(s+1)}, \widehat{\boldsymbol{\Omega}}) \right] & (29)\\ \le&\sum_{i=1}^{n} \log \left\{ \mathbb{E}_{c_i \mid \boldsymbol{a}_i, \widehat{\boldsymbol{\Omega}}, \boldsymbol{e}^{(s)}} \left[P(A_{ij} \mid c_i, e_j^{(s+1)}, \widehat{\boldsymbol{\Omega}}) \right] \right\}. & (30) \end{aligned}$$

We can also formally prove $\ell_{\mathrm{PL}}(\widehat{\boldsymbol{\Omega}}, \boldsymbol{e}^{(s+1)}; \{\boldsymbol{a}_i\}) \ge \ell_{\mathrm{PL}}(\widehat{\boldsymbol{\Omega}}, \boldsymbol{e}^{(s)}; \{\boldsymbol{a}_i\})$. To simplify the notation, we write

$$\begin{aligned} \hat{\tau}_{ik}^{(s)} &= P(c_i = k \mid \boldsymbol{a}_i, \widehat{\boldsymbol{\Omega}}, \boldsymbol{e}^{(s)}) & (31)\\ &= \frac{\hat{\pi}_k \prod_{j=1}^{n} \hat{P}_{ke_j^{(s)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s)}}\right)^{1-A_{ij}}}{\sum_{k=1}^{K} \hat{\pi}_k \prod_{j=1}^{n} \hat{P}_{ke_j^{(s)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s)}}\right)^{1-A_{ij}}}. & (32) \end{aligned}$$

Then we have

$$\ell_{\mathrm{PL}}(\widehat{\boldsymbol{\Omega}}, \boldsymbol{e}^{(s+1)}; \{\boldsymbol{a}_i\}) - \ell_{\mathrm{PL}}(\widehat{\boldsymbol{\Omega}}, \boldsymbol{e}^{(s)}; \{\boldsymbol{a}_i\}) \tag{33}$$

$$= \sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} \hat{\pi}_k \prod_{j=1}^{n} \hat{P}_{ke_j^{(s+1)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s+1)}}\right)^{1-A_{ij}}\right\} - \sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} \hat{\pi}_k \prod_{j=1}^{n} \hat{P}_{ke_j^{(s)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s)}}\right)^{1-A_{ij}}\right\} \tag{34}$$

$$= \sum_{i=1}^{n} \log\left\{\frac{\sum_{k=1}^{K} \hat{\pi}_k \prod_{j=1}^{n} \hat{P}_{ke_j^{(s+1)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s+1)}}\right)^{1-A_{ij}}}{\sum_{k=1}^{K} \hat{\pi}_k \prod_{j=1}^{n} \hat{P}_{ke_j^{(s)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s)}}\right)^{1-A_{ij}}}\right\} \tag{35}$$

$$= \sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} \frac{\hat{\pi}_k \prod_{j=1}^{n} \hat{P}_{ke_j^{(s+1)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s+1)}}\right)^{1-A_{ij}}}{\hat{\pi}_k \prod_{j=1}^{n} \hat{P}_{ke_j^{(s)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s)}}\right)^{1-A_{ij}}} \frac{\hat{\pi}_k \prod_{j=1}^{n} \hat{P}_{ke_j^{(s)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s)}}\right)^{1-A_{ij}}}{\sum_{k=1}^{K} \hat{\pi}_k \prod_{j=1}^{n} \hat{P}_{ke_j^{(s)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s)}}\right)^{1-A_{ij}}}\right\} \tag{36}$$

$$= \sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} \frac{\hat{\pi}_k \prod_{j=1}^{n} \hat{P}_{ke_j^{(s+1)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s+1)}}\right)^{1-A_{ij}}}{\hat{\pi}_k \prod_{j=1}^{n} \hat{P}_{ke_j^{(s)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s)}}\right)^{1-A_{ij}}} \hat{\tau}_{ik}^{(s)}\right\} \tag{37}$$

$$\geq \sum_{i=1}^{n} \sum_{k=1}^{K} \hat{\tau}_{ik}^{(s)} \log\left\{\frac{\prod_{j=1}^{n} \hat{P}_{ke_j^{(s+1)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s+1)}}\right)^{1-A_{ij}}}{\prod_{j=1}^{n} \hat{P}_{ke_j^{(s)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s)}}\right)^{1-A_{ij}}}\right\} \tag{38}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{j=1}^{n} \hat{\tau}_{ik}^{(s)} \left\{\log\left[\hat{P}_{ke_j^{(s+1)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s+1)}}\right)^{1-A_{ij}}\right] - \log\left[\hat{P}_{ke_j^{(s)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s)}}\right)^{1-A_{ij}}\right]\right\} \tag{39}$$

$$= \sum_{j=1}^{n} \left(\sum_{i=1}^{n} \sum_{k=1}^{K} \hat{\tau}_{ik}^{(s)} \left\{\log\left[\hat{P}_{ke_j^{(s+1)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s+1)}}\right)^{1-A_{ij}}\right] - \log\left[\hat{P}_{ke_j^{(s)}}^{A_{ij}} \left(1 - \hat{P}_{ke_j^{(s)}}\right)^{1-A_{ij}}\right]\right\}\right) \tag{40}$$

$$\geq 0, \tag{41}$$

where the first inequality is due to Jensen's inequality, and the second inequality is due to the updating formula (26).

Although the alternating update algorithm stated above is not guaranteed to maximize the pseduo log-likelihood $\ell_{\mathrm{PL}}(\Omega, \boldsymbol{e}; \{\boldsymbol{a}_i\})$, it ensures a non-negative increment in $\ell_{\mathrm{PL}}(\Omega, \boldsymbol{e}; \{\boldsymbol{a}_i\})$ at each iteration.

2.3 Jin's SCORE Algorithm

Jin (2015) proposed the Spectral Clustering On Ratios-of-Eigenvectors (SCORE) algorithm for fast community detection in the degree-corrected SBM (to be introduced in Section 2.4). Here we discuss the intuition behind SCORE in the context of SBM[3].

We treat $\boldsymbol{c} = (c_1, \ldots, c_n) \in \{1, \ldots, K\}^n$ as unknown model parameters, rather than as random variables. For each node i, we write the column vector $\boldsymbol{e}_i \in \mathbb{R}^K$ as its membership vector where $e_i(k) = I(c_i = k)$ for all $1 \le k \le K$. To write the model in matrix form, we define $\boldsymbol{E} := (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n)^\top = [\boldsymbol{E}_1, \ldots, \boldsymbol{E}_K] \in \mathbb{R}^{n\times K}$, and write $\boldsymbol{\Omega} := \boldsymbol{EPE}^\top = \sum_{i=1}^K \sum_{j=1}^K P_{ij} \boldsymbol{E}_i \boldsymbol{E}_j^\top \in \mathbb{R}^{n\times n}$. All the other notations introduced earlier will be retained. Our objective is to recover $\boldsymbol{c}$ from the observed network $\boldsymbol{A}$.

In SBM, for all edge variables $\{A_{ij}, 1 \le i < j \le n\}$, we have

$$
\begin{aligned}
\mathbb{E}[A_{ij}] &= P(A_{ij} = 1) && (42)\\
&= P_{c_i c_j} && \text{(Treat } \boldsymbol{c} \text{ as model parameters)} \quad (43)\\
&= \boldsymbol{e}_i^\top \boldsymbol{P} \boldsymbol{e}_j && \text{(Definition of } \boldsymbol{e}_i\text{'s)} \quad (44)\\
&= (\boldsymbol{EPE}^\top)(i,j) && \text{(Definition of } \boldsymbol{E}) \quad (45)\\
&= \boldsymbol{\Omega}(i,j). && \text{(Definition of } \boldsymbol{\Omega}) \quad (46)
\end{aligned}
$$

Since we do not allow for self-edges, we have $\mathbb{E}[\boldsymbol{A}] = \boldsymbol{\Omega} - \mathrm{diag}(\boldsymbol{\Omega})$. It follows that

$$
\begin{aligned}
\boldsymbol{A} &= \mathbb{E}[\boldsymbol{A}] + (\boldsymbol{A} - \mathbb{E}[\boldsymbol{A}]) && (47)\\
&= \boldsymbol{\Omega} - \mathrm{diag}(\boldsymbol{\Omega}) + \boldsymbol{W}. && \text{(Define } \boldsymbol{W} = \boldsymbol{A} - \mathbb{E}[\boldsymbol{A}]) \quad (48)
\end{aligned}
$$

Recall that our goal is to recover $\boldsymbol{c}$ from $\boldsymbol{A}$. We ought to find out "which part of the data contains the information" (Tukey (1965)). In this context, we note the following:

- $\boldsymbol{\Omega}$ contains the most direct information of the community labels.
- $\mathrm{diag}(\boldsymbol{\Omega})$ contains negligible information compared to $\boldsymbol{\Omega}$.
- $\boldsymbol{W}$ should be considered as some noise, since the upper triangular part of $\boldsymbol{W}$ consists of independent (but not identical) centered-Bernoulli variables.

We next consider the oracle case where we have access to the main signal matrix $\boldsymbol{\Omega} = \boldsymbol{EPE}^\top = \sum_{i=1}^K \sum_{j=1}^K P_{ij} \boldsymbol{E}_i \boldsymbol{E}_j^\top$.

2.3.1 The Oracle Case

Suppose $\boldsymbol{\eta} \in \mathbb{R}^n$ is an eigenvector of $\boldsymbol{\Omega}$ with eigenvalue λ. We wish to find the connection between $\boldsymbol{\eta}$ and the community label vector $\boldsymbol{c}$.

The analysis starts with

$$
\begin{aligned}
\lambda \boldsymbol{\eta} &= \boldsymbol{\Omega} \boldsymbol{\eta} && (49)\\
&= \sum_{i=1}^K \sum_{j=1}^K P_{ij} \boldsymbol{E}_i \boldsymbol{E}_j^\top \boldsymbol{\eta}. && (50)
\end{aligned}
$$

[3]In fact, Jin (2015) demonstrated both theoretically and empirically that the effect of degree heterogeneity can be mitigated by SCORE. In this note, however, we only consider a simpler model, the SBM, to highlight the underlying principles.

Then, for all $1 \le l \le K$, we have

$$\lambda \boldsymbol{E}_l^\top \boldsymbol{\eta} = \boldsymbol{E}_l^\top \sum_{i=1}^K \sum_{j=1}^K P_{ij} \boldsymbol{E}_i \boldsymbol{E}_j^\top \boldsymbol{\eta} \tag{51}$$

$$= \|\boldsymbol{E}_l\|_2^2 \sum_{j=1}^K P_{lj} \boldsymbol{E}_j^\top \boldsymbol{\eta}. \qquad (\boldsymbol{E} \text{ has orthogonal columns}) \tag{52}$$

We may rewrite the previous line in its matrix form:

$$\operatorname{diag}\left(\|\boldsymbol{E}_1\|_2^2, \ldots, \|\boldsymbol{E}_K\|_2^2\right) \boldsymbol{P} \begin{pmatrix} \boldsymbol{E}_1^\top \boldsymbol{\eta} \\ \vdots \\ \boldsymbol{E}_K^\top \boldsymbol{\eta} \end{pmatrix} = \begin{pmatrix} \|\boldsymbol{E}_1\|_2^2 \sum_{j=1}^K P_{1j} \boldsymbol{E}_j^\top \boldsymbol{\eta} \\ \vdots \\ \|\boldsymbol{E}_K\|_2^2 \sum_{j=1}^K P_{Kj} \boldsymbol{E}_j^\top \boldsymbol{\eta} \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{E}_1^\top \boldsymbol{\eta} \\ \vdots \\ \boldsymbol{E}_K^\top \boldsymbol{\eta} \end{pmatrix}. \tag{53}$$

To simplify the notation, we write $\boldsymbol{D} = \operatorname{diag}\left(\|\boldsymbol{E}_1\|_2^2, \ldots, \|\boldsymbol{E}_K\|_2^2\right)$. It follows that $\left(\boldsymbol{E}_1^\top \boldsymbol{\eta}, \ldots, \boldsymbol{E}_K^\top \boldsymbol{\eta}\right)^\top$ is an eigenvector of $\boldsymbol{DP}$ with eigenvalue λ. Further, if $\boldsymbol{a} \in \mathbb{R}^K$ is an eigenvector of $\boldsymbol{DP}$, then $\boldsymbol{\eta}_{\boldsymbol{a}}$ is an eigenvector of $\boldsymbol{\Omega}$ with the same eigenvalue, where

$$\boldsymbol{\eta}_{\boldsymbol{a}} = \sum_{l=1}^K \frac{a(l)}{\|\boldsymbol{E}_l\|_2} \boldsymbol{E}_l. \tag{54}$$

We assume that $\boldsymbol{DP}$ has eigenvectors $\boldsymbol{a}_k, 1 \le k \le K$. Then the eigenvectors of $\boldsymbol{\Omega}$ are given by[4]

$$\boldsymbol{\eta}_k = \sum_{l=1}^K \frac{a_k(l)}{\|\boldsymbol{E}_l\|_2} \boldsymbol{E}_l, \quad 1 \le k \le K, \tag{55}$$

which implies that, for $1 \le i \le n$, we have

$$\eta_k(i) = \sum_{l=1}^K \frac{a_k(l)}{\|\boldsymbol{E}_l\|_2} E_l(i) \tag{56}$$

$$= \sum_{l=1}^K \frac{a_k(l)}{\|\boldsymbol{E}_l\|_2} I(c_i = l) \tag{57}$$

$$= \frac{a_k(c_i)}{\|\boldsymbol{E}_{c_i}\|_2}. \tag{58}$$

This suggests that $\eta_k(i) = \eta_k(j)$ for all $1 \le k \le K$ if $c_i = c_j$ (i.e., nodes i and j belong to the same community). We may take the ratio[5],

$$R(i,k) = \frac{\eta_{k+1}(i)}{\eta_1(i)}, \quad 1 \le i \le n, 1 \le k \le K-1, \tag{59}$$

$$= \frac{a_{k+1}(c_i)}{a_1(c_i)}, \tag{60}$$

and define the matrix $\boldsymbol{R} \in \mathbb{R}^{n \times (K-1)}$ with its (i,k) entry as $R(i,k)$. We finally conclude that

[4] The matrix $\boldsymbol{\Omega}$ has at most K eigenvalues since $\operatorname{rank}(\boldsymbol{\Omega}) = \operatorname{rank}(\boldsymbol{EPE}^\top) \le \min\{\operatorname{rank}(\boldsymbol{E}), \operatorname{rank}(\boldsymbol{P})\} \le K$.

[5] When using the SBM, it is unnecessary to take the ratio. However, in the degree-corrected SBM, the matrix $\boldsymbol{E}$ should be redefined as $\boldsymbol{E} = (\theta_1 \boldsymbol{e}_1, \ldots, \theta_n \boldsymbol{e}_n)^\top$, where the ratios eliminate the nuisance degree parameters (i.e., θ_i's).

- The matrix $\boldsymbol{R}$ has only K distinct rows.
- For any $1 \leq i \neq j \leq n$ with $c_i = c_j$ (i.e., nodes i and j belong to the same community), the i-th and j-th rows of $\boldsymbol{R}$ are identical.

We are now ready to present the SCORE community detection algorithm in the oracle case:

- Compute the eigenvectors of $\boldsymbol{\Omega}$ and denote them by $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_K$.
- Define the matrix $\boldsymbol{R} \in \mathbb{R}^{n \times (K-1)}$ with its (i,k) entry as $R(i,k) = \eta_{k+1}(i)/\eta_1(i)$.
- Nodes i and j belong to the same community if the i-th and j-th rows of $\boldsymbol{R}$ are identical.

2.3.2 The Real Case

We now consider the real case where $\boldsymbol{A}$, instead of $\boldsymbol{\Omega}$, is observed.

Considering what we have learned from the oracle case, it is natural and logical to write the algorithm as follows:

- Compute the K (unit-norm) leading eigenvectors[6] of $\boldsymbol{A}$ and denote them by $\hat{\boldsymbol{\eta}}_1, \ldots, \hat{\boldsymbol{\eta}}_K$.
- Define the matrix $\hat{\boldsymbol{R}} \in \mathbb{R}^{n \times (K-1)}$ with its (i,k) entry as $\hat{R}(i,k) = \hat{\eta}_{k+1}(i)/\hat{\eta}_1(i)$.
- Apply a clustering algorithm to $\hat{\boldsymbol{R}}$, such as k-means[7].

2.4 Extensions of the SBM

We introduce several extensions of the SBM. In the following paragraphs, all notations introduced previously will be retained and used consistently unless explicitly stated otherwise or there exists a conflict. Throughout, we assume that there are K communities in total.

The mixed-membership SBM model (MMSBM; Airoldi et al. (2008)) allows each node to belong to multiple communities by introducing a weight vector $\boldsymbol{\pi}_i \in \mathbb{R}^K$ for each node i. Conditional on $\{\boldsymbol{\pi}_i, 1 \leq i \leq n\}$, the edge variables $\{A_{ij}, 1 \leq i < j \leq n\}$ are assumed to be independent Bernoulli variables with

$$
\begin{aligned}
\mathbb{E}[A_{ij} \mid \boldsymbol{\pi}_i, \boldsymbol{\pi}_j] &= P(A_{ij} = 1 \mid \boldsymbol{\pi}_i, \boldsymbol{\pi}_j) && (61)\\
&= \sum_{k=1}^{K} \sum_{l=1}^{K} \pi_i(k) P_{kl} \pi_j(l) && (62)\\
&= \boldsymbol{\pi}_i^\top \boldsymbol{P} \boldsymbol{\pi}_j. && (63)
\end{aligned}
$$

It is noteworthy that in the original paper Airoldi et al. (2008), the weight vectors (i.e., $\boldsymbol{\pi}_i$'s) are treated as independent random variables drawn from a Dirichlet distribution denoted by $\mathrm{Dir}(\boldsymbol{\alpha})$. The parameters to be estimated are $\boldsymbol{P}$ and $\boldsymbol{\alpha}$.

[6] By "leading eigenvectors", we are comparing the absolute values of the eigenvalues.

[7] The k-means algorithm is also the choice in the original paper Jin (2015).

The degree-corrected SBM (DCSBM; Karrer and Newman (2011)) accounts for degree heterogeneity by incorporating additional degree parameters. Specifically, conditional on the label vector $\boldsymbol{c}$, DCSBM assumes that the edge variables $\{A_{ij}, 1 \le i < j \le n\}$ are independent Poisson variables with

$$\mathbb{E}\left[A_{ij} \mid \boldsymbol{c}\right] = \theta_i \theta_j \lambda_{c_i c_j}, \tag{64}$$

where $\boldsymbol{\Lambda} = [\lambda_{kl}]$ is a $K \times K$ symmetric matrix and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ is a degree parameter vector.

The Degree-Corrected Mixed-Membership (DCMM) model (Jin et al. (2017); Ji et al. (2022); Jin et al. (2024)) permits both degree heterogeneity and mixed memberships. Specifically, DCMM assumes that each node i is associated with a K-dimensional weight vector $\boldsymbol{\pi}_i \in \mathbb{R}^K$ where for $1 \le k \le K$,

$$\begin{aligned} \pi_i(k) &= \text{the } k\text{-th component of } \boldsymbol{\pi}_i && (65)\\ &= \text{the fractional weight of node } i \text{ on community } k. && (66) \end{aligned}$$

Below is an example from Ji et al. (2022) which helps clarify the meaning of $\boldsymbol{\pi}_i$ in DCMM:

> "Suppose $K = 3$ and we have three communities, each being a primary area in statistics: 'Bayes', 'Biostatistics', and 'Non-parametric'. Suppose for author i, $\boldsymbol{\pi}_i = (0.5, 0.3, 0.2)^\top$. In this case, we think author i has 50%, 30%, and 20% of his research interest or impact in these primary areas, respectively."

DCMM further assumes that the edge variables $\{A_{ij}, 1 \le i < j \le n\}$ are independent Bernoulli variables with

$$\begin{aligned} \mathbb{E}[A_{ij}] &= P(A_{ij} = 1) && (67)\\ &= \theta_i \theta_j \sum_{k=1}^{K} \sum_{l=1}^{K} \pi_i(k) P_{kl} \pi_j(l) && (68)\\ &= \theta_i \theta_j \boldsymbol{\pi}_i^\top \boldsymbol{P} \boldsymbol{\pi}_j. && (69) \end{aligned}$$

Note that in Jin et al. (2017), Ji et al. (2022), and Jin et al. (2024), $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$, $\{\boldsymbol{\pi}_i, 1 \le i \le n\}$, and $\boldsymbol{P}$ are all treated as unknown model parameters to be estimated, rather than as random variables drawn from some distributions.

3 Statistical Text Analysis

Ludwig Wittgenstein, the famous Austrian philosopher, once said,

> "The limits of my language mean the limits of my world."

Given n *documents* written with a vocabulary of p *words*, we denote the sets of all documents and words by $\{d^{(1)}, \ldots, d^{(n)}\}$ and $\{w^{(1)}, \ldots, w^{(p)}\}$, respectively. We assume that there are only

K ($\ll \min\{n, p\}$) *topics* that are discussed by all these documents. We denote the set of all topics by $\{z^{(1)}, \ldots, z^{(K)}\}$.

We write $X \in \mathbb{R}^{p\times n}$ as the *word-document-count* matrix, where $X(j, i)$ is the count of the j-th vocabulary in document i. We denote the word count vector of document i by $x_i = (X(1, i), \ldots, X(p, i))^\top \in \mathbb{R}^p$. Then we have $X = [x_1, \ldots, x_n]$.

Throughout, we consider only *bag-of-words* models, i.e., models that focus on the counts of individual words, while neglecting word order and context.

3.1 Mixture of Unigrams

Nigam et al. (2000) used a topic model that Blei et al. (2003) referred to as the *mixture of unigrams* model[8]. We present the model here in a slightly modified form. For each topic $z^{(k)}$, we write its population word frequency vector as $A_k \in \mathbb{R}^p$. We assume that for all $1 \le i \le n$, document i is generated under some latent topic $z_i \in \{z^{(1)}, \ldots, z^{(K)}\}$, and that

$$P(z_i = z^{(k)}) = w(k), \quad 1 \le k \le K. \tag{70}$$

Note that here $P(z_i = z^{(k)})$ does not vary with i, and each document exhibits exactly one topic[9]. Suppose document i has a total of N_i words. We further assume that

$$x_i \mid z_i = z^{(k)} \sim \text{Multinomial}(N_i, A_k), \quad 1 \le i \le n, \tag{71}$$

that is, once the topic $z_i = z^{(k)}$ is given, the distribution of x_i is multinomial with parameters (N_i, A_k).

We write $w = (w(1), \ldots, w(K))^\top \in \mathbb{R}^K$ to combine the weights of different topics into a vector, which we refer to as the *topic weight vector*. We also refer to $A = [A_1, \ldots, A_K] \in \mathbb{R}^{p\times K}$ as the *topic matrix*. Then we have $X = [x_1, \ldots, x_n]$. The main focus is to estimate $\theta = (A, w)$ from X.

[8]However, according to Ke et al. (2023), unigram models are those only model the counts of single words, neglecting word orders and word context.

[9]These assumptions naturally limit the model's capacity to fit a large collection of documents.

3.1.1 Estimation via EM

Suppose all documents are independent. The log-likelihood function for $\theta = (A, w)$ is then computed as

$$\ell(\theta; X) = \sum_{i=1}^{n} \log P(x_i \mid \theta) \tag{72}$$

$$= \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} P(x_i, z_i = z^{(k)} \mid \theta) \right) \tag{73}$$

$$= \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} P(z_i = z^{(k)} \mid \theta)\ P(x_i \mid z_i = z^{(k)}, \theta) \right) \tag{74}$$

$$= \sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} w(k) \ \prod_{j=1}^{p} (A_k(j))^{X(j,i)} \right], \tag{75}$$

where constants are omitted by convention. It is obvious that optimizing $\ell(\theta; X)$ does not admit a closed-form solution, which calls for effective algorithms.

The well-known Expectation-Maximization (EM) algorithm (Dempster et al. (1977)) is a commonly used approach to fitting latent variable models. The algorithm is an iterative procedure that alternates between two steps: the Expectation (E-step) and the Maximization (M-step). In the E-step, it computes the expected value of the complete-data log-likelihood with respect to the conditional distribution of the latent variables given the observed data and current parameter estimates. In the M-step, it maximizes this expected complete-data log-likelihood with respect to the model parameters.

In the mixture of unigrams model, words and documents can be considered as *observed variables* (also referred to as *manifest variables*), while topics can be viewed as *latent variables.* The log-likelihood function for complete data (i.e., $X, z_1, \ldots, z_n$) is computed as

$$\ell(\theta; X, z_1, \ldots, z_n) = \sum_{i=1}^{n} \log P(x_i, z_i \mid \theta) \tag{76}$$

$$= \sum_{i=1}^{n} [\log P(\text{topic } z_i \mid \text{doc } i) + \log P(x_i \mid \text{topic } z_i, \text{doc } i)] \tag{77}$$

$$= \sum_{i=1}^{n} \left[\log P(\text{topic } z_i \mid \text{doc } i) + \sum_{j=1}^{p} X(j,i) \log P(\text{word } j \mid \text{topic } z_i, \text{doc } i) \right] \tag{78}$$

$$= \sum_{i=1}^{n} \left[\log P(\text{topic } z_i \mid \text{doc } i) + \sum_{j=1}^{p} X(j,i) \log P(\text{word } j \mid \text{topic } z_i) \right] \tag{79}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \left[I(z_i = z^{(k)}) \left(\log w(k) + \sum_{j=1}^{p} X(j,i) \log A_k(j) \right) \right]. \tag{80}$$

The **E-step** begins by computing the posterior distribution of z_i after observing x_i:

$$\begin{aligned} P(z_i = z^{(k)} \mid x_i, \theta) &= \frac{P(x_i \mid z_i = z^{(k)}) P(z_i = z^{(k)})}{\sum_{k=1}^{K} P(x_i \mid z_i = z^{(k)}) P(z_i = z^{(k)})} \\ &= \frac{\text{Multinomial}(x_i \mid N_i, A_k)\ w(k)}{\sum_{l=1}^{K} \text{Multinomial}(x_i \mid N_i, A_l)\ w(l)}, \end{aligned} \tag{81}$$

which follows from the well-known Bayes formula. Then, the so-called Q function is given by

$$Q(\theta; \theta^{(t)}) = \mathbb{E}\left[\ell(\theta; X, z_1, \ldots, z_n) \mid X, \theta^{(t)}\right] \tag{82}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \left[P(z_i = z^{(k)} \mid x_i, \theta^{(t)}) \left(\log w(k) + \sum_{j=1}^{p} X(j, i) \log A_k(j) \right) \right], \tag{83}$$

where $P(z_i = z^{(k)} \mid x_i, \theta^{(t)})$ should be computed using Eq. (81), i.e.,

$$P(z_i = z^{(k)} \mid x_i, \theta^{(t)}) = \frac{\text{Multinomial}(x_i \mid N_i, A_k^{(t)})\ w(k)^{(t)}}{\sum_{l=1}^{K} \text{Multinomial}(x_i \mid N_i, A_l^{(t)})\ w(l)^{(t)}}. \tag{84}$$

The **M-step** attempts to maximize $Q(\theta; \theta^{(t)})$ under the following constraints:

$$\sum_{j=1}^{p} A_k(j) = 1, \quad 1 \le k \le K, \tag{85}$$

$$\sum_{k=1}^{K} w(k) = 1, \tag{86}$$

To solve this constrained optimization problem, we introduce Lagrange multipliers $\{\tau_k\}_{k=1}^{K}$ and ρ, and write the Lagrangian function as

$$\mathcal{L}(\theta, \{\tau_k\}_{k=1}^{K}, \rho; \theta^{(t)}) = Q(\theta; \theta^{(t)}) + \sum_{k=1}^{K} \tau_k \left(1 - \sum_{j=1}^{p} A_k(j)\right) + \rho \left(1 - \sum_{k=1}^{K} w(k)\right). \tag{87}$$

Setting the derivatives (w.r.t. $A_k(j)$ and $w(k)$) to zero gives rise to

$$\sum_{i=1}^{n} P(z_i = z^{(k)} \mid x_i, \theta^{(t)}) X(j, i) \frac{1}{A_k(j)} - \tau_k = 0, \quad 1 \le j \le p, 1 \le k \le K, \tag{88}$$

$$\sum_{i=1}^{n} P(z_i = z^{(k)} \mid x_i, \theta^{(t)}) \frac{1}{w(k)} - \rho = 0, \quad 1 \le k \le K. \tag{89}$$

Thus, the parameter updating formula in the M-step is given by

$$A_k(j)^{(t+1)} = \frac{\sum_{i=1}^{n} P(z_i = z^{(k)} \mid x_i, \theta^{(t)}) X(j, i)}{\sum_{j=1}^{p} \sum_{i=1}^{n} P(z_i = z^{(k)} \mid x_i, \theta^{(t)}) X(j, i)}, \quad 1 \le j \le p, 1 \le k \le K, \tag{90}$$

$$w(k)^{(t+1)} = \frac{\sum_{i=1}^{n} P(z_i = z^{(k)} \mid x_i, \theta^{(t)})}{\sum_{k=1}^{K} \sum_{i=1}^{n} P(z_i = z^{(k)} \mid x_i, \theta^{(t)})}, \quad 1 \le k \le K. \tag{91}$$

3.2 Hofmann's pLSI Model

Hofmann (1999) proposed the well-known probabilistic Latent Semantic Indexing (pLSI) model, in which each document is allowed to associated with multiple topics. We next review Hofmann's pLSI model in a slightly modified fashion.

Recall that the sets of all documents, topics, and words are denoted by $\{d^{(1)}, \dots, d^{(n)}\}$, $\{z^{(1)}, \dots, z^{(K)}\}$, and $\{w^{(1)}, \dots, w^{(p)}\}$, respectively. For each document $d^{(i)}$, the word generation process can be described as follows:

- Choose a topic z according to $P(z = z^{(k)}) = P(z^{(k)} \mid d^{(i)}), 1 \le k \le K$.
- Given the topic z, choose a word w according to $P(w = w^{(j)}) = P(w^{(j)} \mid z), 1 \le j \le p$.
- Repeat the above steps N_i times, where N_i is the number of words in document $d^{(i)}$.

We remark that Hofmann's pLSI model essentially assumes conditional independence between a word and a document given the topic.

The parameters to be estimated are

$$\theta = \left\{ P(z^{(k)} \mid d^{(i)}),\ P(w^{(j)} \mid z^{(k)}) \mid 1 \le i \le n, 1 \le j \le p, 1 \le k \le K \right\}. \tag{92}$$

which is of order $\mathcal{O}(K(n+p))$. Suppose all documents are independent. The log-likelihood function for θ is then computed as

$$\begin{aligned}
\ell(\theta; X) &= \sum_{i=1}^{n} \log P(x_i \mid \theta) && (93)\\
&= \sum_{i=1}^{n} \sum_{j=1}^{p} X(j,i) \log P\left(w^{(j)} \mid d^{(i)}, \theta\right) && (94)\\
&= \sum_{i=1}^{n} \sum_{j=1}^{p} X(j,i) \log \left[\sum_{k=1}^{K} P\left(w^{(j)}, z^{(k)} \mid d^{(i)}, \theta\right) \right] && (95)\\
&= \sum_{i=1}^{n} \sum_{j=1}^{p} X(j,i) \log \left[\sum_{k=1}^{K} P\left(w^{(j)} \mid z^{(k)}, \theta\right) P\left(z^{(k)} \mid d^{(i)}, \theta\right) \right]. && (96)
\end{aligned}$$

3.2.1 An Equivalent Formulation

Recently, Ke et al. (2023) and Ke and Wang (2024) introduced an alternative formulation of pLSI, which is essentially equivalent to the original model and is detailed as follows. Specifically, they assume that

$$x_i \sim \text{Multinomial}(N_i, \Omega_i), \quad 1 \le i \le n, \tag{97}$$

where $\Omega_i \in \mathbb{R}^p$ is the word weight vector for document i. For each document i, they require that the word weight vector Ω_i admits the following decomposition:

$$\Omega_i = \sum_{k=1}^{K} w_i(k) A_k, \tag{98}$$

where $w_i = (w_i(1), \dots, w_i(p))^\top \in \mathbb{R}^p$ combines the weights of document i on different topics into a vector.

They further write the model in the matrix form. They refer to $A = [A_1, \dots, A_K] \in \mathbb{R}^{p\times K}$ and $W = [w_1, \dots w_n] \in \mathbb{R}^{K\times n}$ as the *topic matrix* and the *topic weight matrix*, respectively. It follows immediately that

$$\Omega = AW, \tag{99}$$

where $\Omega = [\Omega_1, \dots, \Omega_n] \in \mathbb{R}^{p\times n}$.

Why Model (97)-(99) Is Equivalent to pLSI. Suppose all documents are independent. The log-likelihood function for $\theta = (A, W)$ is then computed as

$$\begin{aligned} \ell(\theta; X) &= \sum_{i=1}^{n}\sum_{j=1}^{p} X(j,i) \log \Omega_i(j) && (100)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{p} X(j,i) \log \left[\sum_{k=1}^{K} A_k(j)\ w_i(k)\right], && (101) \end{aligned}$$

which results from Eqs. (97) and (98). This log-likelihood function is equivalent to that of the pLSI model (Eq. (96)) under the reparameterization:

$$\begin{aligned} A_k(j) &= P\left(w^{(j)} \mid z^{(k)}, \theta\right), \quad 1 \le j \le p, 1 \le k \le K, && (102)\\ w_i(k) &= P\left(z^{(k)} \mid d^{(i)}, \theta\right), \quad 1 \le i \le n, 1 \le k \le K. && (103) \end{aligned}$$

Anchor Words. When a certain word appears only when a specific topic is being discussed, we may regard this word as an *anchor word* (Arora et al. (2012)) of that topic. For example, the phrase (word) "Feixi Late Night Canteen" is used only when discussing my home institution (topic), the University of Science and Technology of China[10]. Rigorously, we call word j an anchor word of topic k if $A_k(j) \neq 0$ and $A_l(j) = 0$ for all $l \neq k$. This implies that when word j appears in a document, we can immediately infer that this document covers topic k (though it may also cover other topics). A practical use of anchor words is that they allow us to interpret each estimated topic and subsequently assign an appropriate label by its anchor words.

Identifiability Issue. To make Model (97)-(99) identifiable (i.e., for a given Ω, there is a unique pair (A, W) such that $\Omega = AW$ holds), we may require the *anchor-word condition* (i.e, each topic has at least one anchor word). According to Donoho and Stodden (2003) and Ke and Wang (2024), this is almost the necessary condition for identifiability of Model (97)-(99).

3.3 Latent Dirichlet Allocation

The latent Dirichlet allocation (LDA) model (Blei et al. (2003)) can be viewed as a Bayesian version of the pLSI model. In the LDA model, we assume that the topic weight vectors $w_1, \dots, w_n$ are i.i.d. drawn from a Dirichlet distribution with parameter $\alpha = (\alpha_1, \dots, \alpha_K)$, i.e., $w_i, \dots, w_n, \overset{\text{i.i.d.}}{\sim} \mathrm{Dir}(\alpha)$.

[10]This is a joke. Please do not take it seriously.

Dirichlet Distribution. A random vector $\gamma = (\gamma_1, \ldots, \gamma_K)^\top \in \mathbb{R}^K$ is said to be Dirichlet distributed with parameter $\alpha = (\alpha_1, \ldots, \alpha_K)$, i.e., $\gamma \sim \mathrm{Dir}(\alpha)$, if it takes value in a $K-1$-simplex (i.e., $\gamma_j \geq 0$ for all $1 \leq j \leq K$ and $\sum_{j=1}^K \gamma_j = 1$) and its probability density function is given by

$$p(\gamma \mid \alpha) = \frac{\Gamma\left(\sum_{i=1}^K \alpha_i\right)}{\prod_{i=1}^K \Gamma\left(\alpha_i\right)} \prod_{i=1}^K \gamma_i^{\alpha_i - 1}, \tag{104}$$

where $\Gamma(\cdot)$ is the gamma function.

Specificially, for each document i, the generative process of LDA can be described as:

- Choose a topic distribution $w_i \sim \mathrm{Dir}(\alpha)$.
- Repeat the following steps N_i times, where N_i is the number of words in document i:
  - Choose a topic z according to $P(z = z^{(k)} \mid w_i) = w_i(k), 1 \leq k \leq K$.
  - Suppose the chosen topic is $z^{(k)}$, choose a word according to $P(\text{word} = w^{(j)} \mid z = z^{(k)}) = A_k(j), 1 \leq j \leq p$.

Here $A_k = (A_k(1), \ldots, A_k(p))^\top \in \mathbb{R}^p$ is the population word frequency vector for topic $z^{(k)}$, $1 \leq k \leq K$. We refer to $A = [A_1, \ldots, A_K] \in \mathbb{R}^{p \times K}$ and $W = [w_1, \ldots w_n] \in \mathbb{R}^{K \times n}$ as the topic matrix and the topic weight matrix, respectively. Our notations here are consistent with earlier sections.

Given the parameters A and α, the joint distribution of x_i and w_i is given by

$$\begin{aligned}
P(x_i, w_i \mid A, \alpha) &= P(w_i \mid \alpha) P(x_i \mid w_i, A) &(105)\\
&= \mathrm{Dir}(w_i \mid \alpha) \prod_{j=1}^p \left(P(\text{word} = w^{(j)} \mid w_i, A)\right)^{X(j,i)} &(106)\\
&= \mathrm{Dir}(w_i \mid \alpha) \prod_{j=1}^p \left(\sum_{k=1}^K P(\text{word} = w^{(j)}, \text{topic} = z^{(k)} \mid w_i, A)\right)^{X(j,i)} &(107)\\
&= \mathrm{Dir}(w_i \mid \alpha) \prod_{j=1}^p \left(\sum_{k=1}^K P(\text{topic} = z^{(k)} \mid w_i) P(\text{word} = w^{(j)} \mid \text{topic} = z^{(k)}, A)\right)^{X(j,i)} &(108)\\
&= \mathrm{Dir}(w_i \mid \alpha) \prod_{j=1}^p \left(\sum_{k=1}^K w_i(k) A_k(j)\right)^{X(j,i)}, &(109)
\end{aligned}$$

where constant factors (w.r.t. A and α) are omitted. Integrating over w_i gives rise to

$$\begin{aligned}
P(x_i \mid A, \alpha) &= \int P(x_i, w_i \mid A, \alpha) \mathrm{d}w_i &(110)\\
&= \int \mathrm{Dir}(w_i \mid \alpha) \prod_{j=1}^p \left(\sum_{k=1}^K w_i(k) A_k(j)\right)^{X(j,i)} \mathrm{d}w_i. &(111)
\end{aligned}$$

Suppose all documents are independent. We take the product of the marginal probabilities of single documents and finally obtain

$$
\begin{aligned}
P(X \mid A, \alpha) &= \prod_{i=1}^{n} P(x_i \mid A, \alpha) &(112)\\
&= \prod_{i=1}^{n} \int \mathrm{Dir}(w_i \mid \alpha) \prod_{j=1}^{p} \left(\sum_{k=1}^{K} w_i(k) A_k(j) \right)^{X(j,i)} \mathrm{d}w_i. &(113)
\end{aligned}
$$

The LDA model can be fitted (i.e., maximizing $P(X \mid A, \alpha)$ w.r.t (A, α)) via either the variational EM algorithm or Gibbs sampling (Porteous et al. (2008)). In this note, we only present the variational EM algorithm.

3.3.1 Estimation via Variational Inference

Suppose Q is some probability density function for W. The starting point of variational EM algorithm is the following derivation:

$$
\begin{aligned}
\log P(X \mid A, \alpha) &= \log P(X \mid A, \alpha) \int Q(W) \mathrm{d}W &(114)\\
&= \int \log P(X \mid A, \alpha)\; Q(W) \mathrm{d}W &(115)\\
&= \mathbb{E}_{Q(W)} \left[\log P(X \mid A, \alpha) \right] &(116)\\
&= \mathbb{E}_{Q(W)} \left[\log \frac{P(X, W \mid A, \alpha)}{P(W \mid X, A, \alpha)} \right] &(117)\\
&= \mathbb{E}_{Q(W)} \left[\log \frac{P(X, W \mid A, \alpha)\; Q(W)}{P(W \mid X, A, \alpha)\; Q(W)} \right] &(118)\\
&= \mathbb{E}_{Q(W)} \left[\log \frac{P(X, W \mid A, \alpha)}{Q(W)} \right] + \mathbb{E}_{Q(W)} \left[\log \frac{Q(W)}{P(W \mid X, A, \alpha)} \right] &(119)\\
&= \mathbb{E}_{Q(W)} \left[\log \frac{P(X, W \mid A, \alpha)}{Q(W)} \right] + D_{\mathrm{KL}} \left(Q(W) \| P(W \mid X, A, \alpha) \right) &(120)\\
&\geq \mathbb{E}_{Q(W)} \left[\log \frac{P(X, W \mid A, \alpha)}{Q(W)} \right], &(121)
\end{aligned}
$$

where the last line follows from the fact that the Kullback-Leibler (KL) divergence is always non-negative (Kullback and Leibler (1951)). The term in the last line is often referred to as the Evidence Lower BOund (ELBO), i.e.,

$$
\begin{aligned}
\mathrm{ELBO}(Q, A, \alpha) &= \mathbb{E}_{Q(W)} \left[\log \frac{P(X, W \mid A, \alpha)}{Q(W)} \right] &(122)\\
&= \mathbb{E}_{Q(W)} \left[\log P(X, W \mid A, \alpha) \right] - \mathbb{E}_{Q(W)} \left[\log Q(W) \right]. &(123)
\end{aligned}
$$

We can clearly observe from Eq. (120) that the log-likelihood $\log P(X \mid A, \alpha)$ is equal to the ELBO term $\mathrm{ELBO}(Q, A, \alpha)$ plus the KL divergence term $D_{\mathrm{KL}} \left(Q(W) \| P(W \mid X, A, \alpha) \right)$. This observation motivates us to maximize $\mathrm{ELBO}(Q, A, \alpha)$, as an alternative to directly maximizing $\log P(X \mid A, \alpha)$, which is the basic idea behind variational inference (Jordan et al. (1999); Wainwright et al. (2008)).

Formally, the variational EM algorithm can be stated as follows:

- **E-step**: Fix (A, α), and solve for $Q = \arg\max_Q \mathrm{ELBO}(Q, A, \alpha)$.
- **M-step**: Fix Q, and solve for $(A, \alpha) = \arg\max_{(A,\alpha)} \mathrm{ELBO}(Q, A, \alpha)$.

These two steps are repeated until $\mathrm{ELBO}(Q, A, \alpha)$ converges.

We introduce the so-called *mean-field* family, i.e,

$$Q(W) = \prod_{i=1}^{n} q_i(w_i). \tag{124}$$

In the LDA model, we may simplify the computation by requiring

$$q_i(w_i) = \mathrm{Dir}(w_i \mid \gamma_i), \quad 1 \le i \le n. \tag{125}$$

Here each $\gamma_i = (\gamma_i(1), \ldots, \gamma_i(K))^\top$ is a K-dimensional vector with $\gamma_i(k) > 0$, $1 \le k \le K$.

Equipped with all these assumptions and tools, we write

$$\begin{aligned}
\mathrm{ELBO}(Q, A, \alpha) &= \mathbb{E}_{Q(W)}\left[\log P(X, W \mid A, \alpha)\right] - \mathbb{E}_{Q(W)}\left[\log Q(W)\right] &(126)\\
&= \sum_{i=1}^{n} \mathbb{E}_{q_i(w_i)}\left[\log P(x_i, w_i \mid A, \alpha)\right] - \sum_{i=1}^{n} \mathbb{E}_{q_i(w_i)}\left[\log q_i(w_i)\right] &(127)\\
&= \sum_{i=1}^{n} \mathbb{E}_{q_i(w_i)}\left\{\log\left[\mathrm{Dir}(w_i \mid \alpha) \prod_{j=1}^{p}\left(\sum_{k=1}^{K} w_i(k) A_k(j)\right)^{X(j,i)}\right]\right\} &(128)\\
&\quad - \sum_{i=1}^{n} \mathbb{E}_{q_i(w_i)}\left[\log \mathrm{Dir}(w_i \mid \gamma_i)\right]. &(129)\\
&= \sum_{i=1}^{n} \mathbb{E}_{q_i(w_i)}\left[\log \mathrm{Dir}(w_i \mid \alpha) + \sum_{j=1}^{p} X(j,i) \log\left(\sum_{k=1}^{K} w_i(k) A_k(j)\right)\right] &(130)\\
&\quad - \sum_{i=1}^{n} \mathbb{E}_{q_i(w_i)}\left[\log \mathrm{Dir}(w_i \mid \gamma_i)\right]. &(131)\\
&= \sum_{i=1}^{n} \mathbb{E}_{q_i(w_i)}\left[\log \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} w_i(k)^{\alpha_k - 1} + \sum_{j=1}^{p} X(j,i) \log\left(\sum_{k=1}^{K} w_i(k) A_k(j)\right)\right] &(132)\\
&\quad - \sum_{i=1}^{n} \mathbb{E}_{q_i(w_i)}\left[\log \frac{\Gamma\left(\sum_{k=1}^{K} \gamma_i(k)\right)}{\prod_{k=1}^{K} \Gamma(\gamma_i(k))} \prod_{k=1}^{K} w_i(k)^{\gamma_i(k) - 1}\right]. &(133)
\end{aligned}$$

4 Deep Learning

Exposure to Deep Learning. During my junior year of college, I enrolled in a course titled Introduction to Deep Learning[11], earning a 98 in the class. Through hands-on projects, I gained valuable experience implementing convolutional neural networks, recurrent neural networks, graphical neural networks, and fine-tuning large language models. Furthermore, during my

[11] An elective course for statistics undergraduates taught by the School of Artificial Intelligence and Data Science at USTC.

research internship at UM, I participated in a seminar on diffusion models led by Professor Jeffrey Regier and Professor Yang Chen, where we discussed recent developments in diffusion models from both theoretical and empirical perspectives.

The Success of Deep Learning. It is generally accepted that "the price to pay for achieving low bias is high variance" (Geman et al. (1992)), a principle commonly referred to as the *bias-variance tradeoff*. From a statistical perspective, deep learning models are over-parameterized, with the number of parameters (weights and biases) extremely larger than the number of samples. Such complex models typically suffer from large variances, leading to poor performance on test sets. In my view, the success of deep learning can be attributed to the following two key factors:

- **The Arrival of Big Data.** The huge sample size, which is commonly encountered in the era of big data, reduces the variance to an acceptable level. This is analogous to the fact that the variance of a kernel estimator at a given point is $\mathcal{O}(\frac{1}{nh})$, where n is the sample size and $h > 0$ is the bandwidth (see, e.g., Proposition 1.1 in Tsybakov (2009)).
- **Modern Computing Power.** It is typically challenging to optimize the loss function for an over-parameterized model. However, advancements in modern computing technologies (e.g., GPUs, TPUs) have made such optimization feasible.

4.1 Diffusion Models

The main purpose of the denoising diffusion probability model (DDPM; Ho et al. (2020)) is to generate data from a desired distribution, especially for image data. As early as 2015, Sohl-Dickstein et al. (2015) introduced the idea of using a Markov chain of transitions between latent states. They referred to the encoder as the forward process, and the decoder as the backward process (Figure 1).

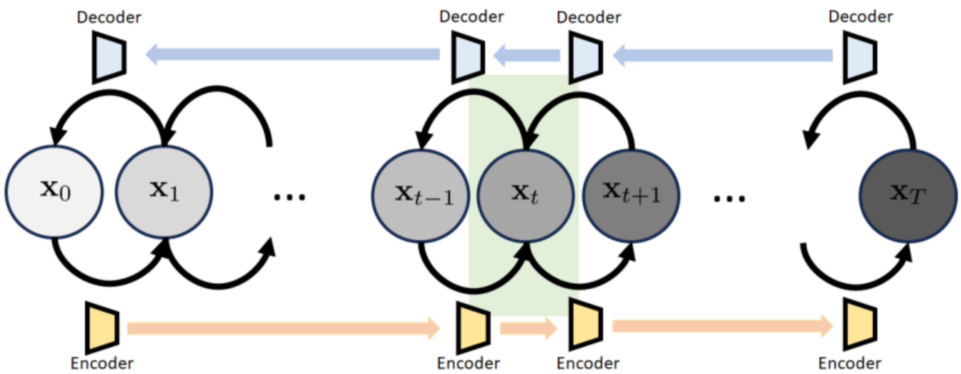

Figure 1: A variational diffusion model by Kingma et al. (2021). The input image is $\mathbf{x}_0$ and the white noise is $\mathbf{x}_T$. The intermediate states $\mathbf{x}_1, \ldots, \mathbf{x}_{T-1}$ are latent variables.

Comparison with VAE. The variational autoencoder (VAE; Kingma (2013)) is an earlier tool for image generation. In a VAE, the latent variable denoted by $\mathbf{z}$, is typically assumed to be standard gaussian distributed, i.e., $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The input vector (typically an image) is denoted by $\mathbf{x}$. The central spirit of an VAE is to learn the conditional distributions $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ associated with the encoder and decoder, respectively, with these two conditional distributions parameterized by separate neural networks. The VAE is, intuitively speaking, a one-step transition process. However, DDPM incrementally updates the latent variables where the assembly of the whole forms the encoder-decoder structure. Other distinctions between DDPMs and VAEs include:

- **Latent Dimension.** In a DDPM, the dimensionality of all latent states is the same as that of the input $\mathbf{x}_0$. In a VAE, the latent variable $\mathbf{z}$ typically has a smaller dimensionality than the input $\mathbf{x}$, because we want the latent low-dimensional variable $\mathbf{z}$ to capture the essential information required to describe $\mathbf{x}$.

- **Forward and Backward Processes.** In a DDPM, the forward process admits a closed-form scheme in that the conditional distribution of $\mathbf{x}_t$ given $\mathbf{x}_{t-1}$, denoted by $q_\phi(\mathbf{x}_t|\mathbf{x}_{t-1})$, is given by $\mathcal{N}(\mathbf{x}_t \mid \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})$. However, in a VAE, the conditional distributions $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ are simple gaussian distributions parameterized by neural networks (which must be learned).

4.1.1 Theoretical Foundation

Despite the huge empirical success, the theory of diffusion models is still in its infancy. Vincent (2011) revealed that the training of denoising networks essentially learns the Stein's score function denoted by $\boldsymbol{s}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$, where $p(\cdot)$ is the probability density function. Therefore, diffusion models fall into the category of score-based generative models (Song et al. (2020)). A recent and highly cited (> 100 on Google Scholar) paper Chen et al. (2023), utilized this insight and demonstrated that

- The Stein's score function can be approximated by a neural network, when the input $\boldsymbol{x}$ admits a low-dimensional linear representation $\boldsymbol{x} = A\boldsymbol{z}$.

- Distribution estimation guarantee (Theorem 3 therein) can be derived using the learned score estimator.

Briefly speaking, Chen et al. (2023) provides insights into distribution recovery from a score-based sampling perspective. Specifically, the sample complexities depend on the intrinsic dimension of $\boldsymbol{x}$ (i.e., the dimensionality of $\boldsymbol{z}$), and are free from the curse of ambient dimensionality. Despite its great value, I think this paper has the following potential drawbacks:

- **Assumptions on the Data.** The input $\boldsymbol{x} = A\boldsymbol{z}$ is assumed to be noise-free, which is impractical in a real-data scenario.

- **Diffusion Models.** The pipeline consists of (1) approximating the Stein's score function by a deep neural network, and (2) generating new data using the discretized backward process. This approach is more like Langevin dynamics (a score-based sampling method;

see, e.g., Section 3 in Chan (2024)), which differs from how data is generated using diffusion models.

- **Lack of Simulation Studies.** The paper does not include any simulation studies to support its results.

5 Statistics in the New Era

Can Deep Learning Beat Statistics? Based on my personal experience, deep learning methods are powerful, but not as powerful as many people claim to be. Deep learning does not easily outperform traditional statistical methods in many real-world scenarios. For example, during my research internship at UM, I found that the performance of a vanilla linear regression model was comparable to that of a neural network when predicting the energy released by a solar flare event. More surprisingly, a standard logistic regression model slightly outperformed the neural network when classifying flare events into pre-defined categories. From a statistical perspective, neural networks are essentially over-parameterized models. When the sample size is not sufficiently large, we cannot expect such complex models to beat classical statistical models.
Challenges and Opportunities. Nowadays, many data are collected through automated processes (automatically, rather than manually) and across different generations of technology[12]. As a result, the quality of the data is often low and measurement errors are inevitable, which calls for effective statistical inference across different scientific contexts. In addition, there is often a gap between the findings published in statistics journals and the concerns of practitioners. In the era of Big Data and AI, bridging this gap becomes even more essential. Future statisticians should have the ability to conduct end-to-end research, from data collection and preprocessing to model development, goodness-of-fit assessments, performance evaluation, and ultimately, communicating findings in ways that are impactful for real-world applications.
Statistics Ph.D. Students in the New Era. With the huge empirical success of large language models, diffusion models, and many other advancements in the era of AI, it would be unwise to ignore them. In my view, future statistics Ph.D. students should actively engage with other communities, including data science, machine learning, and computer science. We should join their seminars, exchange ideas during discussions, stay informed about their recent advancements, and contribute to their journals. Here is a quote from Thomas Henry Huxley:

> "Try to learn something about everything and everything about something."

[12]For example, during my research internship at UM, I learned that the detection algorithms for flare events have evolved over the years.

References

Edo M Airoldi, David Blei, Stephen Fienberg, and Eric Xing. Mixed membership stochastic blockmodels. *Advances in neural information processing systems*, 21, 2008.

Sanjeev Arora, Rong Ge, and Ankur Moitra. Learning topic models–going beyond svd. In *2012 IEEE 53rd annual symposium on foundations of computer science*, pages 1–10. IEEE, 2012.

Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–225, 1974.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

George EP Box and Norman R Draper. *Empirical model-building and response surfaces.* John Wiley & Sons, 1987.

Stanley H Chan. Tutorial on diffusion models for imaging and vision. *arXiv preprint arXiv:2403.18103*, 2024.

Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *International Conference on Machine Learning*, pages 4672–4712. PMLR, 2023.

Corinna Cortes. Support-vector networks. *Machine Learning*, 1995.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? *Advances in neural information processing systems*, 16, 2003.

Howard A Garcia. Temperature and emission measure from goes soft x-ray measurements. *Solar Physics*, 154:275–308, 1994.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

T Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Pengsheng Ji, Jiashun Jin, Zheng Tracy Ke, and Wanshan Li. Co-citation and co-authorship networks of statisticians. *Journal of Business & Economic Statistics*, 40(2):469–485, 2022.

Zhenbang Jiao, Hu Sun, Xiantong Wang, Ward Manchester, Tamas Gombosi, Alfred Hero, and Yang Chen. Solar flare intensity prediction with machine learning models. *Space weather*, 18 (7):e2020SW002440, 2020.

Jiashun Jin. Fast community detection by score. *The Annals of Statistics*, 2015.

Jiashun Jin, Zheng Tracy Ke, and Shengming Luo. Estimating network memberships by simplex vertex hunting. *arXiv preprint arXiv:1708.07852*, 12, 2017.

Jiashun Jin, Zheng Tracy Ke, and Shengming Luo. Mixed membership estimation for social networks. *Journal of Econometrics*, 239(2):105369, 2024.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 83(1):016107, 2011.

Zheng Tracy Ke and Minzhe Wang. Using svd for topic modeling. *Journal of the American Statistical Association*, 119(545):434–449, 2024.

Zheng Tracy Ke, Pengsheng Ji, Jiashun Jin, and Wanshan Li. Recent advances in text analysis. *Annual Review of Statistics and Its Application*, 11, 2023.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39:103–134, 2000.

Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577, 2008.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

John W Tukey. Which part of the sample contains the information? *Proceedings of the National Academy of Sciences*, 53(1):127–134, 1965.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Jiangzhou Wang, Jingfei Zhang, Binghui Liu, Ji Zhu, and Jianhua Guo. Fast network community detection with profile-pseudo likelihood methods. *Journal of the American Statistical Association*, 118(542):1359–1372, 2023.

Linshanshan Wang, Xuan Wang, Katherine P Liao, and Tianxi Cai. Semisupervised transfer learning for evaluation of model classification performance. *Biometrics*, 80(1):ujae002, 2024.

Frank Woodcock. The evaluation of yes/no forecasts for scientific and administrative purposes. *Monthly Weather Review*, 104(10):1209–1214, 1976.